

TRANSMITTAL LETTER TO THE UNITED STATES
DESIGNATED/ELECTED OFFICE (DO/EO/US)
CONCERNING A FILING UNDER 35 U.S.C. 371

T3264-907331

U.S. APPLICATION NO. (IF KNOWN, SEE 37 CFR

09/890211

INTERNATIONAL APPLICATION NO.
PCT/FR00/03231INTERNATIONAL FILING DATE
21 November 2000PRIORITY DATE CLAIMED
26 November 1999

TITLE OF INVENTION

METHOD FOR PRODUCING CONFIGURATION FILES OF OBJECTS CONTAINED IN A COMPUTER SYSTEM

APPLICANT(S) FOR DO/EO/US

Pierre ANDREI and Hoan BUI-XUAN

JUL 26 2001

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. ☒ This is an express request to begin national examination procedures (35 U.S.C. 371(f)). The submission must include items (5), (6), (9) and (24) indicated below.
4. ☒ The US has been elected by the expiration of 19 months from the priority date (Article 31).
5. ☒ A copy of the International Application as filed (35 U.S.C. 371 (c) (2))
 - a. ☐ is attached hereto (required only if not communicated by the International Bureau).
 - b. ☒ has been communicated by the International Bureau.
 - c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
6. ☒ An English language translation of the International Application as filed (35 U.S.C. 371(c)(2)).
 - a. ☒ is attached hereto.
 - b. ☐ has been previously submitted under 35 U.S.C. 154(d)(4).
7. ☐ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371 (c)(3))
 - a. ☐ are attached hereto (required only if not communicated by the International Bureau).
 - b. ☐ have been communicated by the International Bureau.
 - c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
 - d. ☐ have not been made and will not be made.
8. ☐ An English language translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
9. ☒ An oath or declaration of the inventor(s) (35 U.S.C. 371 (c)(4)).
10. ☐ An English language translation of the annexes of the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371 (c)(5)).
11. ☐ A copy of the International Preliminary Examination Report (PCT/IPEA/409).
12. ☒ A copy of the International Search Report (PCT/ISA/210).

Items 13 to 20 below concern document(s) or information included:

13. ☒ An Information Disclosure Statement under 37 CFR 1.97 and 1.98.
14. ☒ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
15. ☒ A **FIRST** preliminary amendment.
16. ☐ A **SECOND** or **SUBSEQUENT** preliminary amendment.
17. ☐ A substitute specification.
18. ☒ A change of power of attorney and/or address letter.
19. ☐ A computer-readable form of the sequence listing in accordance with PCT Rule 13ter.2 and 35 U.S.C. 1.821 - 1.825.
20. ☐ A second copy of the published international application under 35 U.S.C. 154(d)(4).
21. ☐ A second copy of the English language translation of the international application under 35 U.S.C. 154(d)(4).
22. ☐ Certificate of Mailing by Express Mail
23. ☒ Other items or information:

Verification of Translation; Proposed Drawing Corrections & 1 red-lined formal drawing; copies of PCT forms:
PCT/IB/301, 304, 308, First page of Published International Application; PCT/RO/101; copies of IDS cited references;

U.S. APPLICATION NO. OF KNOWN STATE 37 CFR

09/890211

INTERNATIONAL APPLICATION NO.

PCT/FR00/03231

ATTORNEY'S DOCKET NUMBER

T3264-907331

24. The following fees are submitted:

BASIC NATIONAL FEE (37 CFR 1.492 (a) (1) - (5)) :

- ☐ Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO and International Search Report not prepared by the EPO or JPO \$1000.00
- ☒ International preliminary examination fee (37 CFR 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO \$860.00
- ☐ International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO \$710.00
- ☐ International preliminary examination fee (37 CFR 1.482) paid to USPTO but all claims did not satisfy provisions of PCT Article 33(1)-(4) \$690.00
- ☐ International preliminary examination fee (37 CFR 1.482) paid to USPTO and all claims satisfied provisions of PCT Article 33(1)-(4) \$100.00

ENTER APPROPRIATE BASIC FEE AMOUNT =

\$860.00

Surcharge of \$130.00 for furnishing the oath or declaration later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492 (e)).

\$0.00

CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE
Total claims	20 - 20 =	0	x \$18.00
Independent claims	2 - 3 =	0	x \$80.00

\$0.00

\$0.00

Multiple Dependent Claims (check if applicable). ☐

\$0.00

TOTAL OF ABOVE CALCULATIONS =

\$860.00

☐ Applicant claims small entity status. (See 37 CFR 1.27). The fees indicated above are reduced by 1/2.

\$0.00

SUBTOTAL =

\$860.00

Processing fee of \$130.00 for furnishing the English translation later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492 (f)).

\$0.00

TOTAL NATIONAL FEE =

\$860.00

Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31) (check if applicable). ☒

\$40.00

TOTAL FEES ENCLOSED =

\$900.00

Amount to be:

refunded

\$

charged

\$

- a. ☒ A check in the amount of \$900 to cover the above fees is enclosed.
- b. ☐ Please charge my Deposit Account No. _____ in the amount of _____ to cover the above fees. A duplicate copy of this sheet is enclosed.
- c. ☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 50-1165. A duplicate copy of this sheet is enclosed.
- d. ☐ Fees are to be charged to a credit card. **WARNING:** Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

Edward J. Kondracki
MILES & STOCKBRIDGE P.C.
Suite 500
1751 Pinnacle Drive
McLean, VA 22102-3833

SIGNATURE

Edward J. Kondracki

NAME

20,604

REGISTRATION NUMBER

July 26, 2001

DATE

T3264-907331-US3863/PG(PCT)

IN THE UNITED STATES DESIGNATED/ELECTED OFFICE (D.O./E.O./US)

Applicants: Pierre ANDREI and Hoan BUI-XUAN

International
Application No.: PCT/FR00/03231

International
Filing Date: 21 November 2000

U.S. Serial No.: To be Assigned

U.S. Filing Date: July 26, 2001

For: **METHOD FOR PRODUCING CONFIGURATION FILES
OF OBJECTS CONTAINED IN A COMPUTER SYSTEM**

McLean, Virginia

PRELIMINARY AMENDMENT

Honorable Commissioner of Patents
and Trademarks
Washington, D.C. 20231

Sir:

Please amend the subject application, filed concurrently herewith, as
indicated below:

IN THE SPECIFICATION:

Page 14, after line 33, insert the following new paragraph:

09890211.072601

--While this invention has been described in conjunction with specific embodiments thereof, it is evident that many alternatives, modifications and variations will be apparent to those skilled in the art. Accordingly, the preferred embodiments of the invention as set forth herein, are intended to be illustrative, not limiting. Various changes may be made without departing from the true spirit and full scope of the invention as set forth herein and defined in the claims.—

09890211.072601

IN THE CLAIMS:

Please cancel claims 1 – 12 in their entirety and insert the following
new claims:

09990211-072601
T3264-907331-US3863/PG(PCT)

1 --13. A method for creating, in a computer system, at least one
2 configuration file for at least one hardware and/or software object including
3 parameters, said configuration file being written using a descriptive
4 metalanguage whose format is independent of the hardware and/or software to
5 be configured, said configuration file including all or some of the parameters of
6 said object and being based on a description file defining constraints to be
7 obeyed on its structure and its syntax during its writing, comprising
8 - expanding the description file by at least one model comprising at least
9 one parameter written in the description file,
10 - and extending at least some of the parameters of said one model,
11 said steps of expanding and extending being done prior to the writing of the
12 configuration file.

1 14. A method according to claim 13, further comprising using a part of
2 the model comprising the extended parameters as a common factor during the
3 writing of the configuration file and limiting writing of the configuration file to the
4 extension of the parameters not having a value.

1 15. A method according to claim 13, further comprising grouping the
2 objects based on the same description file and then identifying the parameters
3 whose value is identical in all of these objects during the creation of the model,
4 and extending the identified parameters in order to create a common factor in
5 said model.

1 16. A method according to claim 13, further comprising using a
2 common factor and extending only the remaining parameters of an element
3 model, as many times as there are objects based on said element model during
4 the writing of the configuration file, if at least two objects are based on the same
5 model.

1 17. A method according to claim 13, characterized in that the language
2 is extensible and further comprising giving a name for identifying the model in the
3 description file, including in the model a reference of the description file, and said
4 reference defining the constraints to be obeyed on the structure and the syntax
5 of said model.

1 18. A method according to claim 13, characterized in that the language
2 is extensible and further comprising inserting into a model two key words
3 DEFINE and DEFINED, indicating whether a parameter is to be defined
4 (DEFINE) or has been defined (DEFINED) in said model.

1 19. A method according to claim 13, characterized in that the language
2 is the XML language, and further comprising taking as a parameter an element
3 and/or an attribute of an object.

1 20. A method according to claim 19, further comprising expanding the
2 description file by at least one element model comprising at least one parameter
3 (element and/or attribute) described in the description file, and extending all or
4 some of the parameters of said model.

1 21. A method according to claim 19, further comprising giving a name
2 for identifying the element model in the description file, and including in the
3 model a reference to an element of the description file, said reference defining
4 the constraints to be obeyed on the structure and the syntax of said model.

1 22. A method according to claim 19, further comprising including in an
2 element model at least one element model.

1 23. A method according to claim 19, further comprising, at the request
2 of an application using the configuration file, transmitting the common factor and
3 the blocks resulting from the extension of the undefined elements.

1 24. A method according to claim 14, further comprising grouping the
2 objects based on the same description file and then identifying the parameters
3 whose value is identical in all of these objects during the creation of the model,
4 and extending the identified parameters in order to create a common factor in
5 said module.

1 25. A method according to claim 14, further comprising using a
2 common factor and extending only the remaining parameters of an element
3 model, as many times as there are objects based on said element model during
4 the writing of the configuration file, if at least two objects are based on the same
5 model.

1 26. A method according to claim 14, characterized in that the language
2 is extensible and further comprising giving a name for identifying the model in the
3 description file, including in the model a reference of the description file, and said
4 reference defining the constraints to be obeyed on the structure and the syntax
5 of said model.

1 27. A method according to claim 14, characterized in that the language
2 is extensible and further comprising inserting into a model two key words
3 DEFINE and DEFINED, indicating whether a parameter is to be defined
4 (DEFINE) or has been defined (DEFINED) in said model.

1 28. A method according to claim 19, further comprising giving a name
2 for identifying the element model in the description file, and including in the
3 model a reference to an element of the description file, said reference defining
4 the constraints to be obeyed on the structure and the syntax of said model.

1 29. A method according to claim 19, further comprising including in an
2 element model at least one element model.

1 30. A method according to claim 19, further comprising, at the request
2 of an application using the configuration file, transmitting the common factor and
3 the blocks resulting from the extension of the undefined elements.

1 31. A configuration file for at least one hardware and/or software object
2 comprising parameters, said configuration file being written using a descriptive

3 metalanguage whose format is independent of the hardware and/or software to
4 be configured, said configuration file including at least some of the parameters of
5 said object and being based on a description file defining constraints to be
6 obeyed on its structure and its syntax during its writing, characterized in that the
7 description file is expanded, and in that the expansion comprises at least one
8 model having at least one extended parameter written in the description file.

1 32. A configuration file as set forth in claim 31, wherein only the extension
2 of parameters not having a value are included.--

T3264-907331-US3863

IN THE ABSTRACT:

Please delete the Abstract at page 18 in its entirety and substitute the following new Abstract.

09890241-072604

-- ABSTRACT

The object of the invention is to create at least one configuration file for at least one hardware and/or software object comprising parameters. The configuration file is written using a descriptive metalanguage whose format is independent of the hardware and/or software to be configured. The configuration file includes all or some of the parameters of the object and is based on a description file defining constraints to be obeyed on its structure and its syntax during its writing. Prior to the writing of the configuration file, the description file is expanded by at least one model comprising at least one parameter described in the description file, and all or some of the parameters of the model are extended.--

103207369

REMARKS

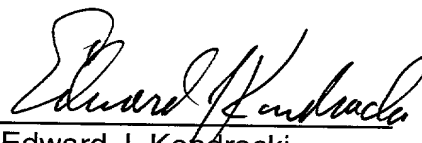
This Preliminary Amendment is filed to eliminate the use of multiple dependent claims, and to correct informalities in the specification, claims and abstract resulting from a literal translation of the French text.

Early action on the merits is earnestly solicited.

Respectfully submitted,

MILES & STOCKBRIDGE P.C.

Date: July 26, 2001

By: 
Edward J. Kondracki
Registration No. 20,604

1751 Pinnacle Drive – Suite 500
McLean, VA 22102-3833
Tel.: 703/903-9000
Fax: 703/610-8686

1 / PRTS

09/890211

JG17 Rec'd PCT/PTO 26 JUL 2001

**METHOD FOR PRODUCING CONFIGURATION FILES OF OBJECTS
CONTAINED IN A COMPUTER SYSTEM**

SPECIFICATION

5

Technical Field

The present invention relates to a method for creating configuration files for objects belonging to a computer system.

The computer system comprises hardware objects (machines, etc.) and/or software
10 objects (applications, etc.). This system may or may not be distributed, or heterogeneous.

The objects comprise parameters contained in a configuration file. The invention applies to any configuration file written in an extensible markup language, i.e. a language that presents information framed by flags. The configuration file is written using a descriptive metalanguage whose format is independent from the hardware and/or software
15 to be configured. A metalanguage is generally defined as a language used to describe another language. The markup language XML (eXtensible Markup Language), known to one skilled in the art, is well adapted to the implementation of the present solution. Let us recall that the specification of the XML language is defined by the consortium W3C (World Wide Web Consortium). This consortium is an organization for the promotion of
20 the World Wide Web, which develops free and open standards and protocols, with concern for maximal interoperability. The XML configuration file has a structure declared in a description file. This description file comprises the description of an object's configuration parameters and is generally referred to as Document Type Definition (DTD). This definition adheres to a particular formalism, also defined in the XML
25 specification of the W3C consortium.

The Prior Art

By definition, the writing of a configuration file in an XML language must obey certain syntactical rules. In essence, an XML document has a logical structure. It is
30 composed of descriptions, elements, comments, character calls and processing instructions, which are indicated in the document by means of a specific flagging. The elements are framed by opening flags, for example <preface> and closing flags, for example </preface>. The elements are structured and describe the parameters of the

09890211.072601

objects. Parameters, like an attribute of an object, can even be included in the flags. For example, it is possible to write <BOOK subject = k>, indicating that the subject attribute of the element BOOK has the value K.

In our exemplary embodiment, the parameters of the objects are defined in a configuration file written in an XML language as defined above.

The main problem is that the objects to be configured number in the thousands, and several of these objects can be based on the same DTD description file and can have identical values for all or some of their parameters. In order to construct the configuration file, the administrator of the management system must therefore extend the parameters described in the description file as many times as there are objects based on this DTD file. More precisely, let's assume that two objects B1 and B2 are based on the same description file (DTD). To configure the object B1, the user must extend all the parameters described in the DTD file. To configure the object B2, the user must again extend all the parameters described in the DTD file. Consequently, the parameters with the same value are written as many times as there are objects based on the same description file. The writing of a configuration file therefore entails redundancies. The XML language being a low-level configuration language, a user is required to write into the configuration file repetitive descriptions of parameters whose semantics, in certain cases, do not correspond to his needs, which requires substantial knowledge on his part of all of the syntaxes offered by the XML language. Because of this, the manufacturer must provide precise documentation with the description file. It is clear that writing a configuration file is extremely time-consuming.

Another problem, linked to the large number of objects to be described, is that if a user on any machine in the network wants to display resources of the computer system configured in the configuration file, the management system must transmit the configuration file to the remote machine through the network. When the configuration file has a large volume, the flow of data between the management system and the client application can be very high and can result in saturating the communication system between the client applications and the management system.

Finally, another problem is that the syntax defined in the recommendations of the W3C consortium must be obeyed to the letter throughout the writing of the configuration file. There is there for a constant risk of error during the writing of a configuration file for an administrator of the management system.

Summary of the Invention

A first object of the invention is to considerably simplify the writing of configuration files, consequently reducing both the time cost of writing them and the risk of write errors.

A second object is to reduce the size of the configuration file.

To this end, the subject of the invention is a method for creating at least one configuration file for hardware and/or software objects present in a computer system, said configuration file being written using a descriptive metalanguage whose format is independent of the hardware and/or software to be configured, this configuration file including all or some of the parameters of said objects and being based on a description file defining constraints to be obeyed on the structure and the syntax during the writing of said configuration file, characterized in that it consists of expanding the description file by at least one model comprising at least one parameter described in the description file, and in that it consists of extending all or some of the parameters of this model.

It also results in a configuration file for hardware and/or software objects present in a computer system, said configuration file being written using a descriptive metalanguage whose format is independent of the hardware and/or software to be configured, this configuration file including all or some of the parameters of said objects and being based on a description file defining constraints to be obeyed on the structure and the syntax during the writing of said configuration file, characterized in that the description file is expanded, in that the expansion comprises at least one model including at least one parameter included in the description file, and in that some of the parameters of this model are extended.

The invention will be better understood through the reading of the following description given as an example and written in reference to the attached drawings.

Description of an Exemplary Embodiment

In the drawings:

- Fig. 1 is a block diagram of the architecture of a computer system to which the invention can be applied;
- Fig. 2 is a view of a model according to the present invention.

Fig. 1 represents a distributed computer system SYS that illustrates a preferred exemplary embodiment of the invention. In the example illustrated, this system SYS includes a management system SG and at least one machine M1. The management system SG comprises at least one operating system, at least one memory for storing information and at least one processor that controls the process of processing information. The term management is used in order to comply with the AFNOR (French Association for Standardization) translation of "gestion." A machine management system such as "Open Master" (registered trademark of BULL S.A.), known to one skilled in the art, is particularly well adapted to the implementation of the invention. This management system is comparable to a set of services that interact with one another to provide an object representation of the real world specifically constituted by the machines of the computer system. It is an object representation that an administrator manipulates (monitoring, action) in order to manage the real world. The object representation involves virtual objects of the real world and constitutes an object model. In other words, an object managed by the management system is an abstract view, defined for the purpose of managing a physical resource (disk, processor, memory, etc.) and/or logical resources (files, processes, semaphores, etc.) of the computer system.

The management system and the machines it manages constitute a Client/Server architecture. In an architecture of this type, a client application interrogates the management system in order to learn the state of the objects managed by the management system. The Client/Server mode has the advantage of allowing a user called a client (or client application) located on a machine, for example using a simple microcomputer or a workstation, to assign part of its job or its operations to be performed to the server, i.e. the management system. In this way, the client has at its disposal a computing capacity that is much larger than that of its microcomputer.

The computer system may be heterogeneous. In order to mask the heterogeneity of the computer system, the management system SG and the machines managed by the management system comprise at least one respective agent associated with a management protocol. An agent performs, among other things, a protocol conversion.

The management system is linked to a managed machine via any network. The network can be a LAN (Local Area Network), or a WAN (Wide Area Network). A set of software layers is interposed between the management system SG and the network RES,

and between the network and each machine. In order to simplify the description, this set of software layers is not represented in Fig. 1.

Each managed object comprises parameters defined in a configuration file, preferably written in a descriptive markup language comprising a structure and including all or some of the parameters (name, at least one attribute, at least one action, etc.) of the objects. The configuration file is based on a separate file called a description file, which defines the constraints on the structure and the syntactical constraints on the parameters for the writing of said configuration file associated with an object of the machine.

Preferably, the configuration file is written in an XML type language, and the description file is a DTD type description file, known to one skilled in the art. In our exemplary embodiment, this XML configuration file and the DTD description file are included in the management system SG.

In our exemplary embodiment, the DTD description file is centralized in the management system so as to be usable by all the machines of the network. Preferably, the managed objects can be represented by a tree, each node of the tree representing a managed object. A display application APV included in the machine M1 can interrogate the configuration file in order to receive the parameters of the configured objects and to display these objects on this machine. Preferably, in order to display the configuration parameters of the objects, the machine M1 is equipped with a standard browser called "Internet Explorer," known to one skilled in the art, in which a program in JAVA language is executed in order to read the configuration file, thus accessing its content and its structure, and in order to transmit the information read to the display application(s) APV.

Description of an object's configuration parameters:

An object comprises as parameters at least one attribute and properties. For example an attribute ID is the identifier of the object, another attribute TYPE designates its type, and another attribute OWNER.

In our example, an object also comprises as parameters:

- the name of the object,
- the actions that can be executed on this object, including the action open for opening the node, the action close for closing the node, the action develop for displaying the nodes subordinate to a node,

- and graphical properties of this node, including the character font type, the address of the icon with which it is associated, and the desired background color.

Thus, in this DTD description file, a "node" element is associated with a node, and there are elements that are subordinate to it and that are respectively associated with:

- the name of the object,
- the actions (open, close, develop) that can be executed on this object,
- and the graphical properties (character font, icon, background color) of this object.

Attributes can be associated with an element. In our exemplary embodiment, the "node" element includes three attributes, namely:

- an attribute ID designating its identifier,
- an attribute TYPE designating its type,
- and an attribute designating its owner.

A DTD description file defining an object of the tree can be written in the following way, obeying the particular formalism defined in the XML specification of the W3C consortium:

```

< !ELEMENT          node          (nodeName,          nodeActions,
nodeGraphicalProperty)>
< !ELEMENT nodeActions (action*)>
< !ELEMENT nodeName (groupName, addressName, versionName)>
< !ATTLIST node Id   CDATA #REQUIRED
                Type CDATA #REQUIRED
                Owner CDATA #REQUIRED
>
< !ELEMENT nodegraphicalproperty (font, icon, color)>

```

In this file, CDATA # REQUIRED indicates that the attribute in question must be a text block containing characters. Moreover, the entry "action*" indicates that the actions do not have any attributes and that the syntax of these actions to be used during the writing of the configuration file is text.

The entry < !ELEMENT nodeName (groupName, addressName, versionName)> indicates that the element "nodeName" comprises three elements (groupName, addressName, versionName) that are subordinate to it. The element "groupName" designates the name of the object, the element
5 "addressName" designates the address of the object, and the element "versionName" designates the version of the object.

This description file will hereinafter correspond to the description file initially created.

Note that the number of parameters in our exemplary embodiment has been
10 reduced in order to simplify the description. In general, a description file comprises a much larger number of parameters.

In the example illustrated, let's assume for example that three objects (OBJ1, OBJ2 and OBJ3) are based on the same DTD description file defined above. The main problem is that in order to build the configuration file, the administrator of the
15 management system has to extend the parameters described in the description file as many times as there are objects based on this DTD file. The administrator must therefore complete the configuration file three times. The time cost of this writing is therefore considerable.

For this reason, the invention consists of expanding the description file by at least
20 one model comprising at least one parameter included in the description file, and in that it consists of extending some of the parameters of this element model. In this example, the invention consists of inserting an element model, the writing of which adheres to the properties defined in the description below.

In our exemplary embodiment, we have to define a set of objects wherein the only
25 parameters that vary among them are

- the element "Name" corresponding to the name of the object (OBJ1, OBJ2 and OBJ3, respectively (JAZZ, POP, SOUL),

- and the attribute "ID" corresponding to the identifier of the object (OBJ1, OBJ2 and OBJ3), respectively (123, 142, 162).

30 These two parameters will hereinafter be referred to as undefined. In the exemplary embodiment, the objects (OBJ1, OBJ2 and OBJ3) have a respective name (JAZZ, POP, SOUL) and a respective identifier (123, 142, 162).

In keeping with our opening hypothesis, the other parameters will have the same value for each object (OBJ1, OBJ2 and OBJ3). They will be referred to as defined. Thus,

- the value of the element corresponding to the actions that can be executed is the same for each object (OBJ1, OBJ2 and OBJ3),
- 5 ▪ the value of the element corresponding to the graphical properties is the same for each object (OBJ1, OBJ2 and OBJ3).

Likewise,

- the value of the attribute TYPE designating the type of the object is the same for each object (OBJ1, OBJ2 and OBJ3),
- 10 ▪ and the value of the attribute OWNER designating its owner is the same for each object (OBJ1, OBJ2 and OBJ3).

In order to simplify the description, arbitrary values will be given to the defined parameters. In our example, the value of the element corresponding to the actions that can be executed on each object (OBJ1, OBJ2 and OBJ3) is ACT1 for the "open" command, ACT2 for the "close" command, and ACT3 for the "develop" command. Likewise, the value of the element corresponding to the graphical properties of each object (OBJ1, OBJ2 and OBJ3) is PRO1 for the character font, PRO2 for the icon, and PRO3 for the background color. Finally, the two attributes TYPE and OWNER, for each object (OBJ1, OBJ2 and OBJ3) have the values "snmp" and "operator," respectively.

- 20 The two main steps of the process according to the invention are respectively
- the writing of the DTD description file and of the expansion associated with this file constituted by at least one element model (step 1),
 - and the writing of the configuration file resulting from the extension of the undefined parameters of the element model (step 2).

25

Writing the description file and inserting the element model into a description file:

The first step must be performed while adhering to certain properties. The parameters of those objects whose value is invariable being identified, the invention consists of inserting the element model MODEL into the DTD description file created. The element model is distinguished from the other elements of the description file in the sense that it includes at least one parameter with a value.

First of all, the model, while retaining the written formalism of a DTD description file defined in the XML specification of the W3C consortium, comprises

- a header MODEL with a specific name "tnode"
- and a reference to a defined element "node" of the DTD description file initially created.

The model MODEL can be written in the following way:

```
< !MODEL tnode ELEMENT =node >
```

indicating that the model MODEL has a specific name "tnode" and that it is based on the description of the element "node" previously defined in the description file (DTD).

Secondly, in this model, the undefined elements are written in a particular way. In order to distinguish a defined element from an undefined element in the element model, the elements to be defined are marked in this model by means of a specific flag whose header is, for example, <!DEFINE...>. Moreover, the elements to be defined are identified by a name "tnodeName" and a reference to an element nodeName of the initial description file, indicating the element of the previously defined description file on which the element model "tnodeName" is based. In our example, an element to be defined may be written in the following way:

```
< !DEFINE tnodeName...ELEMENT nodeName >.
```

Unlike the undefined elements, the defined elements of the element model are written in the same way as for the writing of an XML configuration file.

Thirdly, the writing of the undefined attributes adheres to certain properties. In this model, in order to distinguish the attributes to be defined and the defined attributes, the invention consists of inserting two key words DEFINE and DEFINED, indicating whether an attribute parameter is to be defined (DEFINE) or has been defined (DEFINED). In our example, the attribute ID is to be defined during the writing of the configuration file, while the attributes TYPE and OWNER are defined and have the values "snmp" and "operator," respectively. In our example, the list of attributes is relative to the element model MODEL "tnode" and can be written in the following way:

```
< ! ATTLIST tnode
    ID DEFINE
    TYPE DEFINED " snmp "
    OWNER DEFINED " operator "
>
```

Finally, the DTD description file that results from such a configuration can be written in the following way:

```

< !MODEL tnode ELEMENT=node
< !DEFINIR tnodeName ELEMENT=nodeName>
5  < nodeActions >
    <action name=open> ACT1</action>
    <action name=close>ACT2</action>
    <action name=develop>ACT3</action>
    </ nodeActions >
10  < nodeGraphicalproperties >
    <character font PRO1... />
    <lcon>PRO2</lcon>
    < background color PRO3/>
    < / Graphicalproperties >
15  >
    < ! ATTLIST tnode
        ID DEFINE
        TYPE DEFINED " snmp "
        OWNER DEFINED " operator "
20  >.

```

This file constitutes a common factor FC for the writing of the configuration file and describes the parameters (elements and/or attributes) to be defined.

25 **Writing the configuration file**

Fig. 2 is a view of the configuration file that results from the utilization of the element model.

The writing of the configuration file corresponds to the second step. This operation consists of using the element model MODEL defined in the DTD description file and of extending the undefined elements and attributes. During the writing of the
30 configuration file, the invention consists in using the part comprising the extended

parameters as a common factor, and in that the writing is limited to the extension of the parameters not having a value.

In the example, three objects (OBJ1, OBJ2 and OBJ3) having the respective names (JAZZ, POP, SOUL) and a respective identifier (123, 142, 162) are to be
5 configured, and the writing of the configuration file for these three objects is limited to writing the following three blocks (B1, B2 and B3):

(B1)

```
<tnode id= "123">  
  < tnodeName >  
10    <nodeName>  
      <groupName> JAZZ </groupName>  
      <addressName> db0 </ addressName >  
      <versionName> 8.0 <versionName>  
      </nodeName>  
15    </ tnodeName >  
  </tnode>
```

(B2)

```
<tnode id= "142">  
20  < tnodeName >  
    <nodeName>  
      <groupName> POP </groupName>  
      <addressName> db1 </ addressName >  
      <versionName> 8.1 <versionName>  
25    </nodeName>  
    </ tnodeName >  
  </tnode>
```

(B3)

```
30 <tnode id= "162">  
    < tnodeName >  
      <nodeName>
```



```

        <groupName> SOL </groupName>
        <addressName> db2 </ addressName >
        <versionName> 8.2 <versionName>
    </nodeName>
5    </ tnodeName >
</tnode>

```

The invocation number of the element model corresponds to the number of objects based on this model MODEL. These three blocks have as their common factor the one created in the description file.

10 According to a variant, an element model can contain other element models. Thus, in a configuration file, it is possible to use, within an element model reference (for example "tnode") another element model reference. In fact, let us assume that in the DTD description file there is an element model "tOracleNodeName" whose defined elements are:

- 15 ▪ the element named groupName
- and the element versionName

and whose undefined element is addressName. The writing of this element model can be as follows.

```

< !MODELE tOracle8NodeName ELEMENT=nodeName
20    <groupName> Oracle </groupName>
    < !DEFINIR tOracleDb ELEMENT=addressName>
        <versionName> 8.0 <versionName>
>

```

25 This way, during the writing of the configuration file, it is possible to use the model "tOracleNodeName" to define the element nodeName. The configuration file is then written in the following way:

```

<tnode Id= "123">
    < tnodeName >
30    < tOracle8NodeName>
        < tOracleDb>
        <addressName> db0 </addressName>
    </tnodeName>
</tnode>

```

```

        </ tOracleDb>
    </ tOracle8NodeName>
</ tnodeName >
</tnode>

```

5

In a general way, the subject of the invention is a method for creating, in a computer system, at least one configuration file for at least one hardware and/or software object comprising parameters, said configuration file being written using a descriptive metalanguage whose format is independent of the hardware and/or software to be configured, this configuration file including all or some of the parameters of said object and being based on a description file defining constraints to be obeyed on its structure and its syntax during its writing, characterized in that it consists, prior to the writing of the configuration file, of expanding the description file by at least one model comprising at least one parameter described in the description file, and of extending all or some of the parameters of this model.

Then, during the writing of the configuration file, the invention consists of using the part of the model comprising the extended parameters as a common factor, and in that the writing of the configuration file is limited to the extension of the parameters not having a value.

During the creation of the model, we have seen that the invention can consist, first or all, of grouping the objects based on the same description file, then of identifying the parameters whose value is identical in all of these objects, and finally, of extending these parameters in order to create a common factor in this model.

The invention consists, for example, during the writing of the configuration file, if at least two objects are based on the same module, of using the common factor and extending only the remaining parameters of this model, as many times as there are objects based on this element model.

The language used is extensible. In our example, we have seen that the invention consists of giving a name for identifying the model in the description file, and in that it consists of including in the model a reference of the description file, this reference defining the constraints to be obeyed on the structure and the syntax of this model. We have also seen that in our example, the invention consists of inserting into the model two

key words DEFINE and DEFINED, indicating whether a parameter is to be defined (DEFINE) or has been defined (DEFINED) in this model.

Preferably, the language of the configuration file is the XML language, the invention consisting of taking as a parameter an element and/or an attribute of an object.

5 In this variant, the invention consists of expanding the description file by at least one element model comprising at least one parameter (element and/or attribute) described in the description file, and of extending all or some of the parameters of this element model. Likewise, the invention consists of giving a name for identifying the element model in the description file, and consists of including in the model a reference to an element of the
10 description file, this reference defining the constraints to be obeyed on the structure and the syntax of this model.

Finally, we have seen that, at the request of an application using the configuration file, the invention can consist of transmitting the common factor and the blocks resulting from the extension of the undefined elements.

15 Lastly, this results in a configuration file for at least one hardware and/or software object comprising parameters, said configuration file being written using a descriptive metalanguage whose format is independent of the hardware and/or software to be configured, this configuration file including all or some of the parameters of said object and being based on a description file defining constraints to be obeyed on its structure and
20 its syntax during its writing, characterized in that the description file is expanded, and in that the expansion comprises at least one model.

In conclusion, the invention offers many advantages. A first advantage is the considerable simplification of the writing of a configuration file. In essence, the defined parameters being extended in the element model, the writing of the configuration file is
25 limited to the extension of the undefined parameters. Consequently, the administrator thus avoids the repetition of the parameters that are identical among the objects. The time cost of writing and the risk of write errors during the writing of the configuration file is greatly reduced. Moreover, the insertion of a common factor into the model considerably reduces the size of the configuration file. Advantageously, at the request of a client application,
30 the management system transmits the configuration file that includes the common factor of the element model and, for each of the objects, the parameters of this element model that have been extended during the writing of the configuration file. The size of the configuration file being reduced, the transfer of this file occurs faster.

CLAIMS

1 1. Method for creating, in a computer system, at least one configuration file
2 for at least one hardware and/or software object comprising parameters, said
3 configuration file being written using a descriptive metalanguage whose format is
4 independent of the hardware and/or software to be configured, this configuration file
5 including all or some of the parameters of said object and being based on a description
6 file defining constraints to be obeyed on its structure and its syntax during its writing,
7 characterized in that it consists, prior to the writing of the configuration file,
8 - of expanding the description file by at least one model comprising at least one
9 parameter written in the description file,
10 - and of extending all or some of the parameters of this model.

1 2. Method according to claim 1, characterized in that it consists, during the
2 writing of the configuration file, of using the part of the model comprising the extended
3 parameters as a common factor, and in that the writing of the configuration file is limited
4 to the extension of the parameters not having a value.

1 3. Method according to claim 1 or 2, characterized in that it consists, during
2 the creation of the model, of grouping the objects based on the same description file and
3 then identifying the parameters whose value is identical in all of these objects, and in that
4 it consists of extending these parameters in order to create a common factor in this
5 module.

1 4. Method according to any of claims 1 through 3, characterized in that it
2 consists, during the writing of the configuration file, if at least two objects are based on
3 the same model, of using the common factor and extending only the remaining
4 parameters of this model, as many times as there are objects based on this element model.

1 5. Method according to any of claims 1 through 4, characterized in that the
2 language is extensible and in that it consists of giving a name for identifying the model in
3 the description file, and in that it consists of including in the model a reference of the
4 description file, and in that it consists of including in the model a reference of the

5 description file, this reference defining the constraints to be obeyed on the structure and
6 the syntax of this model.

1 6. Method according to any of claims 1 through 5, characterized in that the
2 language is extensible and in that it consists of inserting into a model two key words
3 DEFINE and DEFINED, indicating whether a parameter is to be defined (DEFINE) or
4 has been defined (DEFINED) in this model.

1 7. Method according to any of claims 1 through 6, characterized in that the
2 language is the XML language, and in that it consists of taking as a parameter an element
3 and/or an attribute of an object.

1 8. Method according to claim 7, characterized in that it consists of expanding
2 the description file by at least one element model comprising at least one parameter
3 (element and/or attribute) described in the description file, and of extending all or some of
4 the parameters of this model.

1 9. Method according to claim 7 or 8, characterized in that it consists of giving
2 a name for identifying the element model in the description file, and in that it consists of
3 including in the model a reference to an element of the description file, this reference
4 defining the constraints to be obeyed on the structure and the syntax of this model.

1 10. Method according to any of claims 7 through 9, characterized in that it
2 consists of including in an element model at least one element model.

1 11. Method according to any of claims 7 through 10, characterized in that it
2 consists, at the request of an application using the configuration file, of transmitting the
3 common factor and the blocks resulting from the extension of the undefined elements.

1 12. Configuration file for at least one hardware and/or software object
2 comprising parameters, said configuration file being written using a descriptive
3 metalanguage whose format is independent of the hardware and/or software to be
4 configured, this configuration file including all or some of the parameters of said object

5 and being based on a description file defining constraints to be obeyed on its structure and
6 its syntax during its writing, characterized in that the description file is expanded, and in
7 that the expansion comprises at least one model as defined in any of claims 1 through 11.

0989024.072604

ABSTRACT

The object of the invention is to create at least one configuration file for at least one hardware and/or software object comprising parameters, said configuration file being
5 written using a descriptive metalanguage whose format is independent of the hardware and/or software to be configured, this configuration file including all or some of the parameters of said object and being based on a description file defining constraints to be obeyed on its structure and its syntax during its writing. The invention consists, prior to the writing of the configuration file,

- 10 - of expanding the description file by at least one model comprising at least one parameter described in the description file,
 - and of extending all or some of the parameters of this model.

Fig. 2

#9151144

0989024.07601

~~1/1~~

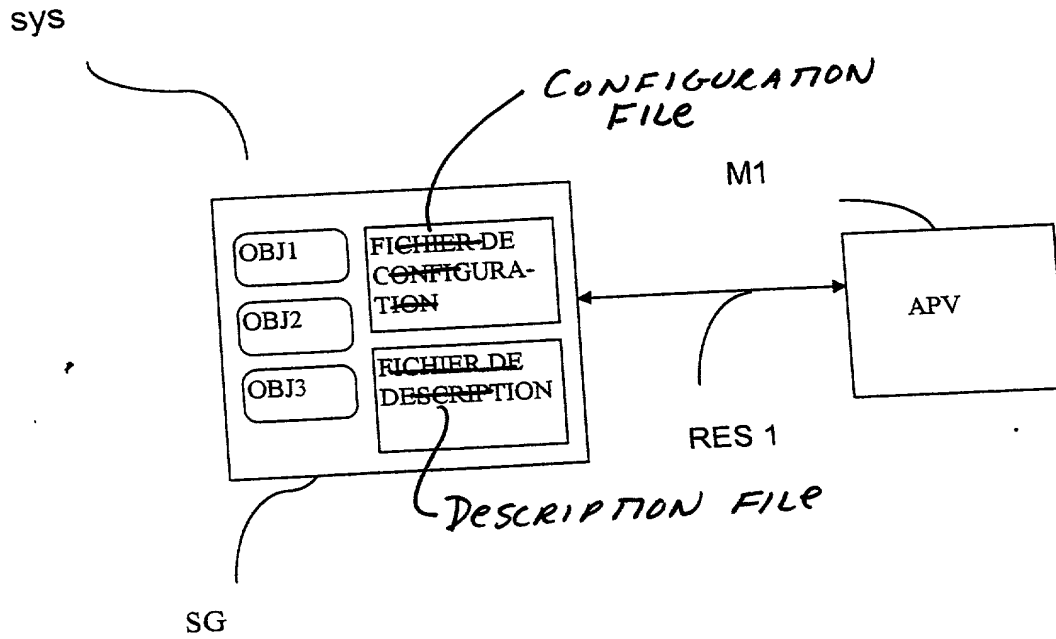


Figure 1

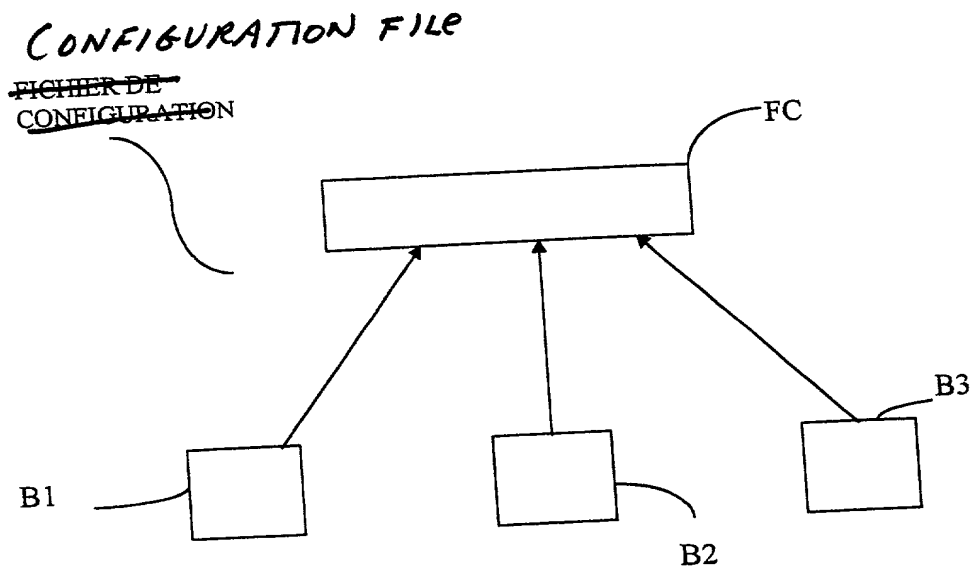


Figure 2

09890211.072601

Declaration and Power of Attorney For Patent Application

Declaration Pour Demandes de Brevets Avec Pouvoirs

French Language Declaration

En tant qu'inventeur nommé ci-après, Je déclare par le présent acte que:

Mon nom, mon domicile, mon adresse postale, ma nationalité sont ceux qui figurent ci-après,

Je déclare que je crois être l'inventeur original, premier et unique (si un seul nom figure sur le présent acte) ou un des co-inventeurs, originaux et premiers (si plusieurs noms figurent sur le présent acte) du sujet revendiqué et pour lequel un brevet est demandé sur la base de l'invention intitulée:

Procédé de création de fichiers de configuration d'objets inclus dans un système informatique

dont la description
(cocher la case correspondante)

☒ est annexée au présent acte.

☐ a été déposée _____

Numéro de série de la demande _____

et modifiée le _____
(si approprié)

Je déclare par le présent acte avoir examiné et compris le contenu de la description identifiée ci-dessus, revendications y compris, et le cas échéant telle que modifiée par l'amendement cité plus haut.

Je reconnais le devoir de divulguer l'information qui est en rapport avec l'examen de cette demande selon Titre 37 du Code des Règlements Fédéraux §1.56(a).

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

the specification of which
(check one)

☐ is attached hereto.

☐ was filed on _____ as

Application Serial No. _____

and was amended on _____
(if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

French Language Declaration

Je revendique par le présent acte le bénéfice de priorité étrangère selon Titre 35, du Code des Etats-Unis, §119 de toute demande de brevet ou d'attestation d'inventeur énumérée ci-après, et j'ai identifié également ci-après toute demande étrangère de brevet ou d'attestation d'inventeur ayant une date de dépôt antérieure à celle de la demande pour laquelle la priorité est revendiquée.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior foreign applications

Priority claimed

Demande(s) de brevet antérieure(s) dans un autre pays:

Droit de priorité revendiqué

<u>FR 9914882</u>	<u>France</u>	<u>26 11 1999</u>
(Number) (Numéro)	(Country) (Pays)	(Day/Month/Year Filed) (Jour/Mois/Année de dépôt)

<input checked="" type="checkbox"/> Yes Oui	<input type="checkbox"/> No Non
---	---

<u> </u>	<u> </u>	<u> </u>
(Number) (Numéro)	(Country) (Pays)	(Day/Month/Year Filed) (Jour/Mois/Année de dépôt)

<input type="checkbox"/> Yes Oui	<input type="checkbox"/> No Non
--	---

<u> </u>	<u> </u>	<u> </u>
(Number) (Numéro)	(Country) (Pays)	(Day/Month/Year Filed) (Jour/Mois/Année de dépôt)

<input type="checkbox"/> Yes Oui	<input type="checkbox"/> No Non
--	---

Je revendique par le présent acte, le bénéfice selon Titre 35 du Code des Etats-Unis, §120 de toute(s) demande(s) américaines énumérée(s) ci-après et, dans la mesure où le sujet de chacune des revendications de cette demande n'est pas divulgué dans la demande américaine antérieure, de la façon définie par le premier paragraphe de Titre 35 du Code des Etats-Unis, §112, je reconnais le devoir de divulguer l'information pertinente selon Titre 37 du Code des Règlements Fédéraux, §1.56(a), toute information qui se présente entre la date de dépôt de la demande antérieure et la date de dépôt de la demande, soit nationale, soit internationale PCT.

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Serial No.)
(No. de Demande)

(Filing Date)
(Date de Dépôt)

(Etat)
(brevetée, pendante,
abandonné)

(Status)
(patented, pending,
abandoned)

(Application Serial No.)
(No. de Demande)

(Filing Date)
(Date de Dépôt)

(Etat)
(brevetée, pendante,
abandonnée)

(Status)
(patented, pending,
abandoned)

Je déclare par le présent acte que toutes mes déclarations, à ma connaissance, sont vraies et que toutes les déclarations faites à partir de renseignements ou de suppositions, sont tenues pour être vraies; de plus, toutes ces déclarations ont été faites en sachant que de fausses déclarations volontaires ou autres actes de même nature sont sanctionnées par une amende ou un emprisonnement, ou les deux, selon la Section 1001, du Titre 18 de Code des Etats-Unis et que de telles déclarations délibérément fausses peuvent compromettre la validité de la demande ou du brevet délivré.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

French Language Declaration

POUVOIR: En tant qu'inventeur, je désigne l'(les) avocat(s) et/ou l'(les) agent(s) suivant(s) pour poursuivre la procédure de cette demande et traiter toute affaire la concernant supris du Bureau des Brevets et de Marques:

Harold L. Stowell, Reg. 17,233
 Edward J. Kondracki, Reg. 20,604
 Dennis P. Clarke, Reg. 22,549
 William L. Feeney, Reg. 29,918
 John C. Kerins, Reg. 32,421

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. (list name and registration number)

Harold L. Stowell, Reg. 17,233
 Edward J. Kondracki, Reg. 20,604
 Dennis P. Clarke, Reg. 22,549
 William L. Feeney, Reg. 29,918
 John C. Kerins, Reg. 32,421

Adresser toute correspondance à:

Edward J. Kondracki, Esq.
KERKAM, STOWELL, KONDRACKI
& CLARKE, P.C.
5203 Leesburg Pike, Suite 600
Falls Church, VA 22041

Send Correspondence to:

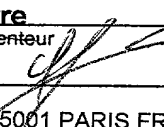
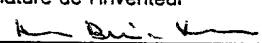
Edward J. Kondracki, Esq.
KERKAM, STOWELL, KONDRACKI
& CLARKE, P.C.
5203 Leesburg Pike, Suite 600
Falls Church, VA 22041

Adresser toute communication téléphonique à:
 (Nom) (Numéro de téléphone)

Edward J. Kondracki, Esq.
(703) 998-3302

Direct Telephone Calls to: (name and telephone number)

Edward J. Kondracki, Esq.
(703) 998-3302

Nom complet du seul ou premier inventeur		Full name of sole or first inventor	
ANDREI Pierre			
Signature de l'inventeur	Date	Inventor's signature	Date
	<u>28 décembre 1999</u>		
Domicile		Residence	
<u>10, rue Hérold 75001 PARIS FRANCE</u> <u>FRX</u>			
Nationalité		Citizenship	
<u>Française</u>			
Adresse Postale		Post Office Address	
<u>10, rue Hérold 75001 PARIS FRANCE</u>			
Nom complet du second co-inventeur, le cas échéant		Full name of second joint inventor, if any	
BUI-XUAN Hoan			
Signature de l'inventeur	Date	Second Inventor's signature	Date
	<u>2 décembre 1999</u>		
Domicile		Residence	
<u>1 avenue Gambetta 75020 PARIS FRANCE</u> <u>FRX</u>			
Nationalité		Citizenship	
<u>Française</u>			
Adresse Postale		Post Office Address	
<u>1 avenue Gambetta 75020 PARIS FRANCE</u>			

(Fournir les mêmes renseignements et la signature de tout co-inventeur supplémentaire.)

(Supply similar information and signature for third and subsequent joint inventors.)